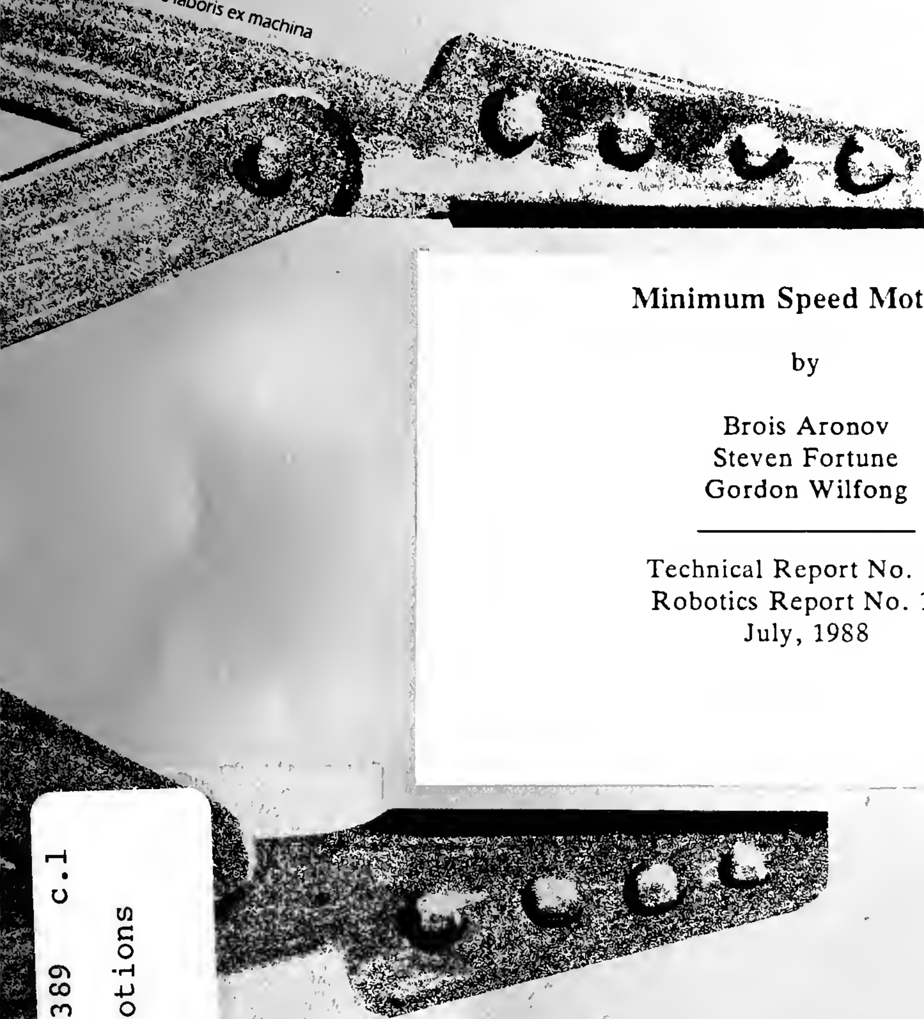
Robotics Research Technical Report

Generatorium omnis laboris ex machina



Minimum Speed Motions

by

Brois Aronov
Steven Fortune
Gordon Wilfong

Technical Report No. 389
Robotics Report No. 163
July, 1988

New York University
ant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012

Minimum Speed Motions

by

Brois Aronov
Steven Fortune
Gordon Wilfong

Technical Report No. 389
Robotics Report No. 163
July, 1988

New York University
Dept. of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012

Work on this paper has been supported by Office of Naval Research Grant N00014-87-K-0129 National Science Foundation CER Grant DCR-83-20085, National Science Foundation Grant subcontract CMU-406349-55586, and by grants from the Digital Equipment Corporation and the IBM Corporation.

Minimum Speed Motions

Boris Aronov (*)

Courant Institute of Mathematical Sciences

Steven Fortune

AT&T Bell Laboratories

Gordon Wilfong

AT&T Bell Laboratories

ABSTRACT

We consider the problem of determining the speed required for an object to avoid moving obstacles in an environment which is fully specified, both in space and in time, by n linear constraints. A solution is presented that, given a full description of the environment and the initial configuration of the system (that is, initial position and starting time of the object), answers in $O(\log n)$ time queries of the form: "What is the lowest speed limit that the object can obey while still being able to reach the query configuration from the initial configuration without colliding with the obstacles?" The algorithm requires $O(n \log n)$ preprocessing time and $O(n)$ space.

July 8, 1988

Minimum Speed Motions

Boris Aronov (*)

Courant Institute of Mathematical Sciences

Steven Fortune

AT&T Bell Laboratories

Gordon Wilfong

AT&T Bell Laboratories

1. Introduction

Until recently, work in the area of algorithmic motion planning has mostly concentrated on the problem of finding a (collision-free) trajectory for a system of objects in a static environment. (For a survey of results and an overview of methods, see [Y].) A common approach to solving such problems is to define *configuration space* in which each point corresponds to a placement of the objects and to search for a path in this space that avoids the points corresponding to placements that result in collisions. In such a model there is no mention of time. As long as the objects move according to the computed path no collision will occur independent of the rate at which the path is followed. Thus while the various objects must move with specific speeds relative to one another in order to follow a given path in configuration space, there remains a degree of freedom in choosing the rate at which the system of objects progresses towards the goal placement.

Introducing time into the model permits the study of more general environments. For example, it makes it possible to model an environment that changes with time. Also, kinematic constraints can be imposed on the objects to be moved. Motion planning in such a model requires not only that we specify the intermediate placements through which the objects must pass in order to get from the initial placement to the final placement, but the time at which the objects reach each intermediate placement must also be determined. That is, a (collision-free) *motion* must be computed. A strategy used in several papers (e.g. [RS], [OD] and [KZ]) is to add a dimension to configuration space to represent time and call this space *time-configuration space*. Let *free space* be the points (t,x) in time-configuration space such that if the system is in the position defined by x at time t then no collision occurs. Then a time-monotonic path in free space defines a collision-free motion for the system. An example of the study of such a model can be found in [RS] where Reif and Sharir consider various motion planning problems in 1, 2 and 3 dimensions where the goal is to compute a motion for an object (whose speed may be bounded) within an environment consisting of moving obstacles. Ó'Dúnlaing [OD] considers the problem of finding a motion for an object confined between two moving barriers in a 1-dimensional space, given an acceleration bound.

Introducing time into the model seems to significantly increase the complexity of the problems. In the model without time, Schwartz and Sharir [SS] and Canny [C] have shown that the *path* planning problem for a system that can be described using only algebraic constraints can be solved in polynomial time provided that the number of degrees of freedom of the system is bounded. On the other hand, it has been shown [CR] that the two-dimensional problem of

(*) The work of the first author was supported in part by an AT&T Bell Laboratories Ph.D. Scholarship. Part of the work was performed at AT&T Bell Laboratories, Murray Hill, NJ.

motion planning with a speed bound is already NP-hard.

Reif and Sharir [RS] observe that finding a speed-bounded motion in the one-dimensional case is relatively easy. In fact, they demonstrate an algorithm that uses a standard sweep-line procedure to decide whether it is possible to get from one configuration (where a configuration specifies a time and a position of the object) to another without exceeding the given speed limit. The algorithm requires $O(n\log n)$ time where n is the complexity of the description of free space. (Kant and Zucker [KZ] describe another algorithm for this problem that requires the construction of the visibility graph (see [Ed, Chap. 12]) of free space and hence can take $O(n^2)$ time.) Reif and Sharir [RS] note that their sweep-line algorithm can be easily modified so that, for a fixed speed limit v, and fixed starting configuration c, the algorithm can answer "Is there a motion from c to c′ with speed bounded by v?" in $O(\log n)$ time for query configuration c' after $O(n\log n)$ preprocessing time.

We consider a natural extension to the problem of [RS] where the speed limit v is also part of the query. That is, each query consists of a configuration c' and a speed bound v and we wish to answer the question, "Is there a motion from c to c′ with speed bounded by v?". In order to answer such a query, it is clearly sufficient to answer the question "What is the maximum speed required in order for the object to be able to reach configuration c' from c?" for query configuration c′. We provide two algorithms for solving this problem: the first algorithm, described in Section 3, is simple to describe and justify. The algorithm is based on the visibility graph of free space. It requires $O(n^2)$ preprocessing time and $O(n^2)$ space in the worst case and takes $O(n)$ time to answer a query. (An output-sensitive version of this algorithm is briefly discussed at the end of Section 3.) The description and especially justification of the second algorithm are more complicated, but it requires only $O(n\log n)$ preprocessing time and $O(n)$ space and guarantees $O(\log n)$ time response to a query. Both algorithms can produce a piecewise-linear motion from the initial to the query configuration in additional time proportional to the number of (maximal) linear segments in the motion produced.

Section 2 contains definitions and a formal model of the problem. Section 3 describes the first algorithm based on a visibility graph approach. Section 4 introduces a geometric structure that facilitates fast query processing, and concludes with an efficient sweep-line procedure for computing it. We discuss various applications, generalizations and open problems in Section 5.

2. Preliminaries

We consider the following model. Let B be an object constrained to move along a one-dimensional track P. Suppose there are moving obstacles that obstruct the motion of B along P at specified times. The goal is to get B from its starting placement somewhere on P at a certain time to some goal placement along P at a prescribed time while avoiding the moving obstacles. We consider the problem of finding such a collision-free motion of B that minimizes the maximum speed necessary to achieve the target placement at the desired time.

As in [KZ], P can be thought of as a solution to a path planning problem for B with respect to some static environment. Then the problem we study deals with the collision-free motions of B along such a path P when moving obstacles are introduced into the environment.

A *configuration* of the system is a pair $z = (t(z), x(z))$ where $t(z)$ is a moment in time and $x(z)$ is some position along P. We say that $x(z)$ is the *position* of B (in configuration z). The set of all configurations is called *time-configuration space* and is represented by the plane with time along the horizontal axis and position along the vertical axis. We say that B *collides* with an obstacle if the interior of B intersects the obstacle. Let T be the set of all configurations (t, x) where B can be at position x at time t without colliding with any obstacle. We consider the case where T is a closed polygonal region of time-configuration space whose boundary consists of n *walls* (maximal line segments or rays). The endpoints of walls are called *corners*. We assume that T is the closure of its interior, exactly two walls meet at a corner and no two corners have the same time coordinate. The connected regions of the complement of T are referred to as *time obstacles*.

A *time-monotonic path* $p:[a,b] \rightarrow T$ is a path such that $t(p(s)) > t(p(s'))$ whenever $s > s'$. Such paths will hereafter be parameterized by time. It is reasonable to assume that B cannot move with infinite speed and so any collision-free motion of B is represented by a time-monotonic path in T. Henceforth whenever we use the term 'path' we shall mean a time-monotonic path in T. The *speed* of a path $p:[a,b] \rightarrow T$, is given by

$$speed(p) = \sup_{a \leq t_1 < t_2 \leq b} \left\{ \frac{|x(p(t_2)) - x(p(t_1))|}{t_2 - t_1} \right\}.$$

If p is a path from z_1 to z_2 and for any other path p' from z_1 to z_2, $speed(p) \leq speed(p')$, then p is called a *minimum speed path*. If $p:[a,b] \rightarrow T$ and $p':[b,c] \rightarrow T$ are two paths with $p(b) = p'(b)$, we use $p \,||\, p'$ to denote the path $P:[a,c] \rightarrow T$ with $P(t) = p(t)$ for $t \in [a,b]$ and $P(t) = p'(t)$ for $t \in [b,c]$. For two configurations $z_1 \neq z_2$ where $t(z_1) < t(z_2)$, let $seg[z_1,z_2]$ denote the closed (directed) line segment connecting z_1 to z_2 in time-configuration space. Similarly let $seg(z_1,z_2)$ be the open line segment from z_1 to z_2 and $seg(z_1,z_2]$ and $seg[z_1,z_2)$ the half-open line segments. We write $speed(z_1,z_2)$ for $speed(seg[z_1,z_2])$, which is simply the absolute value of the slope of $seg[z_1,z_2]$. In this notation, the definition of the speed of an arbitrary path p can be rewritten as $speed(p) = sup\{speed(z_1,z_2)\}$, with the supremum taken over all pairs of configurations z_1, z_2 on p with $t(z_1) < t(z_2)$. The following easily verified property of path speeds is used in subsequent results.

Triangle Inequality: For configurations z_1, z_2, and z_3 with $t(z_1) < t(z_2) < t(z_3)$, $speed(z_1,z_3) \leq \max\{speed(z_1,z_2), speed(z_2,z_3)\}$. The inequality is strict unless z_1, z_2 and z_3 are collinear.

We now use the Triangle Inequality to show the following result.

Lemma 1: For path $p = p_1 \,||\, \cdots \,||\, p_k$, $speed(p) = \max_{1 \leq i \leq k}\{speed(p_i)\}$.

Proof. By definition, $speed(p) \geq \max_{1 \leq i \leq k}\{speed(p_i)\}$. To show the reverse inequality we consider the case where $k = 2$. The general case will follow by a trivial induction. We show that, for any distinct configurations z_1 and z_2 on p, $speed(z_1,z_2) \leq \max\{speed(p_1), speed(p_2)\}$. If for some i, z_1 and z_2 are configurations on the same p_i then it is trivially true. If not, then $t(z_1) < t(z_2)$ implies z_1 is on p_1 and z_2 is on p_2. Let z be the last configuration on p_1 (and hence the first of p_2). The Triangle Inequality implies that

$$speed(z_1,z_2) \leq \max\{speed(z_1,z), speed(z,z_2)\} \leq \max\{speed(p_1), speed(p_2)\}. \ \square$$

A piecewise-linear path p in T between two configurations such that each maximal line segment of p (except possibly for the final and initial segments) connects two corners is called a *canonical path*. We next show that it is sufficient to consider canonical paths when searching for a minimum speed path.

Lemma 2: If there is a path p from configuration z_s to configuration z_f, then there is a canonical path from z_s to z_f whose speed is finite and does not exceed $speed(p)$.

Proof. Let p be a path from z_s to z_f. We first show that p can be transformed into a piecewise-linear path. Since p is time-monotonic it can be broken up into a finite number of maximal subpaths p_i (from configuration z_i to configuration z_{i+1}) with the property that there is no corner c for which $t(z_i) < t(c) < t(z_{i+1})$. Therefore $seg[z_i,z_{i+1}] \subseteq T$. By definition, $speed(p_i) \geq speed(z_i,z_{i+1})$ and so replacing each p_i with $seg[z_i,z_{i+1}]$ results in a piecewise-linear path p' and, by Lemma 1, $speed(p') \leq speed(p)$. Since p' is time-monotonic, all linear segments of p' have finite speed and so by Lemma 1 $speed(p')$ is finite.

Index z_s, z_f and the corners c with $t(z_s) < t(c) < t(z_f)$ so that $t(c_{i+1}) > t(c_i)$ $(i \geq 1)$. For a piecewise-linear path q from z_s to z_f let $k(q)$ be the greatest j such that c_j is on q and the subpath of q from z_s to c_j is canonical. We exhibit a transformation that takes an arbitrary non-canonical piecewise-linear path q and produces a piecewise-linear path q' with $k(q') > k(q)$ and $speed(q') \leq speed(q)$. Starting with p' and applying this transformation a finite number of times results in a canonical path P from z_s to z_f with $c_{k(P)} = z_f$ and $speed(P) \leq speed(p')$, as

asserted. We now describe the required transformation. Let z be the first configuration on q past $c_{k(q)}$ for which $seg(c_{k(q)}, z]$ contains a corner. By continuity, $seg[c_{k(q)}, z] \subseteq T$. Replace the section of q between $c_{k(q)}$ and z by $seg[c_{k(q)}, z]$ (thus increasing $k(q)$) and call the resulting path q'. Again, by Lemma 1 and the definition of the speed of a path, $speed(q') \leq speed(q)$. □

The following result is a simple consequence of Lemma 2 and the fact that there are only a finite number of canonical paths between two given configurations.

Corollary 3: If there is a path from z_o to z_f then there is a minimum speed path from z_o to z_f. Moreover, such a path can be chosen to be canonical.

As a result of Corollary 3, we shall hereafter only consider canonical paths.

By Lemma 1, for a piecewise-linear path $p = seg[z_o, z_1] \,||\, \cdots \,||\, seg[z_{k-1}, z_k]$, $speed(p) = \max_{0 \leq i \leq k-1} \{speed(z_i, z_{i+1})\}$. Throughout the remainder of this paper the initial configuration z_o will be fixed. Let *Reach* be the set of all configurations z in T such that there is a path from z_o to z and let $U = T - Reach$. The configurations in *Reach* are said to be *reachable* and those in U *unreachable*. Let *Illegal* be the complement of *Reach* in time-configuration space. The configurations in *Illegal* are called *illegal configurations*. For configuration $z \in Reach$, $z \neq z_o$, let $min\_speed(z)$ denote the speed of a minimum speed path from z_o to z. Define $min\_speed(z_o) = 0$ and $min\_speed(z) = \infty$ for $z \in U$. Corner c is called a *predecessor* of configuration $z \in Reach$ if there is a minimum speed path p from z_o to z such that the last corner before z on p is c. For configurations z_1 and z_2 with $t(z_1) < t(z_2)$, define

$$cost(z_1, z_2) = \max\{min\_speed(z_1), speed(z_1, z_2)\}.$$

Intuitively, $cost(z_1, z_2)$ would be the minimum speed of any path consisting of a path from z_o to z_1 followed by $seg[z_1, z_2]$ if $seg[z_1, z_2] \subseteq T$. For convenience, we will also set $cost(z, z) = 0$.

Lemma 4: Let $z \in Reach$ and c_i, $1 \leq i \leq m$, be the corners such that $seg[c_i, z] \subseteq T$. Then $min\_speed(z) = \min_{1 \leq i \leq m} \{cost(c_i, z)\}$.

Proof. Follows from Corollary 3 and the definitions of *min_speed* and *cost*. □

Corollary 5: Corner c is a predecessor of $z \in Reach$ if and only if $seg[c, z] \subseteq T$ and $cost(c, z) \leq cost(c', z)$ for any other corner c' with $seg[c', z] \subseteq T$.

3. Visibility graph approach

We now describe a simple algorithm for finding a minimum speed path based on the visibility graph (see, for example, [Ed, Chap. 12]) of T. Given a full description of T, the fixed initial configuration $z_o \in T$ and $O(n^2)$ time for preprocessing, the algorithm will in linear time compute $min\_speed(z)$ for query configuration z and produce a minimum speed path from z_o to z.

Recall that z_o is considered to be the corner of a 0-dimensional time obstacle. Define the *visibility digraph* $D = (V, A)$ of T to be the directed graph where V is the set of corners of T and there is an arc $a = (c_1, c_2)$ in A if $seg[c_1, c_2] \subseteq T$ and $seg(c_1, c_2)$ contains no corners.

The preprocessing step of the algorithm is as follows:

(1) Compute the visibility digraph $D = (V, A)$ of T.

(2) For each corner c of T, compute $min\_speed(c)$ and set $pred(c)$ to be some predecessor of c if $c \in Reach$.

Step (1) of the preprocessing can done in time $O(n^2)$ using the techniques of [AAGHI]. Step (2) can be accomplished by performing the following computations.

(i) For each corner $c \neq z_o$ set $min\_speed(c) = \infty$ and $pred(c) = undefined$. Set $min\_speed(z_o) = 0$ and $pred(z_o) = undefined$.

(ii) Topologically sort D.

(iii) Traverse D in topological order, and as corner c is visited, perform the following for each c' where $(c, c') \in A$:

Let $cost(c,c') = \max\{min\_speed(c), speed(c,c')\}$.
If $min\_speed(c') \geq cost(c,c')$
then set $min\_speed(c') = cost(c,c')$ and $pred(c') = c$.

Since there are $O(n)$ corners and $O(n^2)$ arcs in A, step (i) takes $O(n)$ time, step (ii) can be done in $O(n\log n)$ time (just sort the corners by the time coordinate) and step (iii) takes $O(n^2)$ time in the worst-case. Therefore the entire preprocessing can be done in $O(n^2)$ time.

Lemma 6: Step (2) correctly computes $min\_speed(c)$ and $pred(c)$ for each corner c.

Proof. Follows from Lemma 4 and Corollary 5 by an easy induction on the topological order of the vertices of D. □

For a query configuration z, the preprocessing of [AAGHI] allows us to find in time $O(n)$ the set $Vis(z)$ of all the corners c for which $seg[c,z] \subseteq T$ (and $seg(c,z)$ contains no corners). By Lemma 4,

$$min\_speed(z) = \min_{c \in Vis(z)} \{\max\{min\_speed(c), speed(c,z)\}\}$$

and so computing $min\_speed(z)$ takes $O(n)$ time. A minimum speed path to z can be computed in time $O(n)$ using the *pred* function since a time-monotonic path can pass through each corner at most once.

We also remark that, replacing the visibility graph algorithm of [AAGHI] by the output-sensitive procedure of [KM], we can speed up the above algorithm to use only $O(m+n\log n)$ preprocessing time and $O(m)$ space, where m is the number of arcs in A. (The procedure of [KM] can also be used to answer queries of the form "What obstacle corners can see the query point?" in $O(n)$ time[K].)

Though the approach discussed in this section was easy to describe and its correctness was straight forward to verify, it has two major disadvantages. The practical disadvantage of this solution is that the implementation of step (1) following [AAGHI] would be quite challenging whereas the solution given in the next section employs a relatively standard sweep-line technique and does not rely on such complicated geometric primitives. From a theoretical viewpoint, the approach described in the next section is asymptotically superior in preprocessing time, query processing time, and storage requirements.

4. Sweep-line approach

In this section we describe a sweep-line method for computing a planar partition of the time-configuration space such that each face of the partition either consists of illegal configurations or of reachable configurations with a common predecessor. We describe how this partitioning can be accomplished in $O(n\log n)$ time. It will be shown that the partition consists of $O(n)$ faces and that these faces can be described in linear space. Therefore using standard techniques [ST] we can answer queries of the form "What is a predecessor of configuration z?" in $O(\log n)$ time by determining the face containing z. As the sweep-line passes through corner c we compute $min\_speed(c)$ and a predecessor of c. Thus once we know a predecessor of z we can compute $min\_speed(z)$ in constant time. A minimum speed path can be computed in additional time proportional to the number of corners on it using the *pred* function.

4.1. Partitioning *Reach*

In this section we study the geometry of the partition of *Reach* into regions $R(c)$ such that the latest predecessor of any $z \in R(c)$ is c.

Recall that we assume throughout our discussion that no two corners have equal time coordinates. Suppose c_1 and c_2 are two corners with $t(c_1) < t(c_2)$. Define

$$R(c_1,c_2) = \{z : t(z) > t(c_2) \text{ and } cost(c_1,z) < cost(c_2,z)\} \text{ and}$$

$$R(c_2,c_1) = \{z : t(z) > t(c_2) \text{ and } cost(c_2,z) \leq cost(c_1,z)\}.$$

Let $b(c_1,c_2)$ (and $b(c_2,c_1)$) denote the common boundary of $R(c_1,c_2)$ and $R(c_2,c_1)$ and call

$b(c_1,c_2)$ the *bisector* of c_1 and c_2. By continuity, $cost(c_1,z) = cost(c_2,z)$ for $z \in b(c_1,c_2)$. Moreover, whenever $min\_speed(c_1) \neq min\_speed(c_2)$, the converse holds, i.e., $cost(c_1,z) = cost(c_2,z)$ implies $z \in b(c_1,b_2)$. However, if $min\_speed(c_1) = min\_speed(c_2)$ (see Figure 1) then there could be configurations z in the interior of $R(c_2,c_1)$ (and hence not on $b(c_1,c_2)$) where $cost(c_i,z) = min\_speed(c_i)$, for $i = 1,2$, and so $cost(c_1,z) = cost(c_2,z)$. In Figure 1, $cost(c_1,z) = cost(c_2,z)$ for all $z \in R(c_2,c_1)$ but the bisector $b(c_1,c_2)$ only consists of the two rays out of c_2.

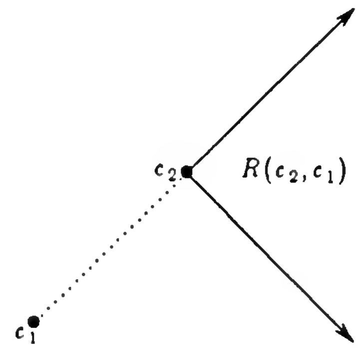


Figure 1. $min\_speed(c_1) = min\_speed(c_2) = speed(c_1,c_2)$

Lemma 7: Let c_1 and c_2 be corners with $t(c_1) < t(c_2)$. The bisector $b(c_1,c_2)$ consists of a constant number of line segments, rays and hyperbolic arcs. It is an empty set, a ray emanating from c_2 away from c_1, or a single continuous curve unbounded at either end.

Proof. Let S be the set of configurations with time coordinate at least $t(c_2)$. Suppose $b(c_1,c_2)$ is nonempty. By definition $b(c_1,c_2) \subseteq S$. The curves $\{z : speed(c_i,z) = min\_speed(c_i)\}$ for $i = 1,2$ divide S into a constant number of two-dimensional open convex cells. The closure of each cell C is such that for each $i = 1,2$ either $cost(c_i,z) = speed(c_i,z)$ for all z in the closure of C or $cost(c_i,z) = min\_speed(c_i)$ for all z in the closure of C. Recall that, $cost(c_1,z) = cost(c_2,z)$ for all $z \in b(c_1,c_2)$. Consider the open cell C where $cost(c_i,z) = min\_speed(c_i)$ for $i = 1,2$. If $min\_speed(c_1) \neq min\_speed(c_2)$ then $b(c_1,c_2) \cap C = \emptyset$ since $z \in C$ implies $cost(c_1,z) \neq cost(c_2,z)$. If $min\_speed(c_1) = min\_speed(c_2)$ then again $b(c_1,c_2) \cap C = \emptyset$ since C is open and all configurations in C must belong to $R(c_2,c_1)$. Therefore for any cell C, either $b(c_1,c_2) \cap C = \emptyset$ or $b(c_1,c_2) \cap C$ is the portion of one of the following curves lying in C:

(i) $\{z : speed(c_i,z) = min\_speed(c_j), i \neq j\}$

(ii) $\{z : speed(c_1,z) = speed(c_2,z)\}$.

A type (i) curve is a union of two rays and a type (ii) curve is the union of a ray and a hyperbolic arc that meet at c_2. A simple analysis shows that the intersection of the hyperbolic arc of a type (ii) curve with a closed cell of the partition is a connected piece. Clearly the intersection of a ray of type (i) or (ii) curve with a closed cell is connected. Therefore $b(c_1,c_2)$ consists of a constant number of line segments, rays and hyperbolic arcs.

It is also easily seen that any ray emanating from c_1 intersects $b(c_1,c_2)$ in at most one connected component and any vertical line intersects it in at most two points. The Lemma follows. □

A consequence of the proof of Lemma 7 is that $b(c_1,c_2)$ can be computed in constant time once $min\_speed(c_1)$ and $min\_speed(c_2)$ are known. Figure 2 is an example of $b(c_1,c_2)$. The dotted lines represent the boundaries of the cells in the above proof and the solid rays and arc form $b(c_1,c_2)$.

For each reachable corner c, let $R(c)$ (called the *region* of c), be the set of reachable

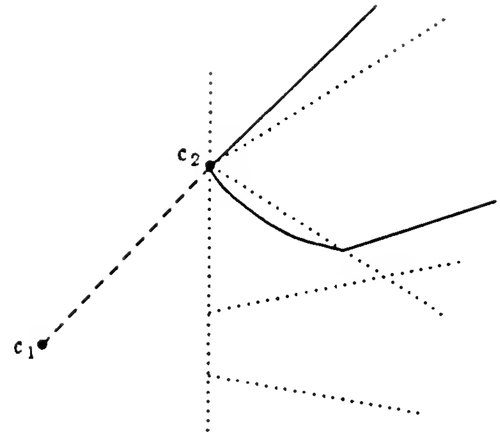


Figure 2. Bisector in case where $min\_speed(c_1) < min\_speed(c_2) < speed(c_1, c_2)$

configurations z such that c is a predecessor of z and, if c' is another predecessor of z, then $t(c') < t(c)$. Note that the regions of distinct corners are disjoint and, in fact, the (non-empty) regions form a partition of *Reach*. Also observe that the line segment connecting corner c to any configuration in $closure(R(c))$ stays completely in T.

We wish to compute a planar map with the property that each of its faces is completely contained in a single region (or completely contained in *Illegal*). Moreover, we require that region membership for configurations on the edges and vertices of such a map be easily determined. The first property is trivially satisfied by defining faces to be connected components of region interiors (or connected components of the interior of *Illegal*). The remaining points will constitute map edges and vertices. A complicating factor in classifying points is that a region need not be closed or open; it also may or may not include two-dimensional areas. To start with, we must determine the general shape of a region, which is the subject of the following lemma.

We say that a set of configurations R in time-configuration space is a *punctured star* with respect to configuration c if, for all $z \in R$, $t(z) > t(c)$ and the line segment $seg(c, z]$ is contained in R.

Lemma 8: For each corner c, $R(c)$ is a punctured star with respect to c.

Proof. By definition of a region, for all $z \in R(c)$, $t(z) > t(c)$. Let $a \in R(c)$ and suppose b is some configuration on the line segment $seg(c, a)$ with $b \in R(c')$ for some $c' \neq c$. Notice that $cost(c, a) = cost(c, b)$. Since $b \in R(c')$, $cost(c', b) \leq cost(c, b)$. Therefore

$$\begin{aligned} cost(c', a) &\leq \max\{cost(c', b), speed(b, a)\} \\ &\leq \max\{cost(c, b), speed(b, a)\} \\ &= cost(c, b) = cost(c, a). \end{aligned}$$

If $t(c) < t(c')$ then $cost(c', a) \leq cost(c, a)$ contradicts the assumption that $a \in R(c)$.

Suppose $t(c') < t(c)$. In this case, $b \in R(c')$ implies $cost(c', b) < cost(c, b)$. Since $a \in R(c)$ and $b \in R(c')$ there are no corners on the open line segments $seg(c, a)$ and $seg(c', b)$. In particular, c, c', b and a are not collinear. Thus there is some $b' \neq b$ on $seg(c', b)$ such that $seg[b', a] \subseteq T$. Since

$$speed(b', b) = speed(c', b) \leq cost(c', b) < cost(c, b),$$

the Triangle Inequality implies that

$$speed(b',a) < \max\{speed(b',b), speed(b,a)\} \leq \max\{cost(c,b), speed(b,a)\} = cost(c,b).$$

Therefore

$$cost(c',a) \leq \max\{cost(c',b'), speed(b',a)\} < cost(c,b) = cost(c,a),$$

again contradicting $a \in R(c)$. □

Lemma 9: The interior of $R(c)$ is a punctured star with respect to c.

Proof. Let z be a configuration in the interior of $R(c)$. Then there is a line segment $seg[a,b]$ contained in $R(c)$, passing through z and perpendicular to $seg[c,z]$. Since $R(c)$ is a punctured star with respect to c, the interior of the triangle cab is contained in $R(c)$. Thus $seg(c,z]$ is in the interior of $R(c)$. □

We now determine (a) what curves form the boundary of a region and (b) region membership for boundary configurations. We begin by characterizing configurations that lie on region boundaries.

Lemma 10: Suppose $z \neq c$ is on the boundary of $R(c)$.

(1) If z is not on the boundary of any other region, then either $t(z) = t(c)$ or z lies on a wall.

(2) If $t(z) = t(c)$, then z is not on the boundary of any other region.

Proof. (1): Suppose z lies on the boundary of $R(c)$, $z \neq c$ and z does not lie on the boundary of any other region. Then $c \in Reach$. If z lies on a wall then there is nothing to prove. So suppose z does not lie on a wall; and for the sake of contradiction suppose $t(z) > t(c)$. Let c' be the corner on $seg[c,z]$ with greatest time coordinate; $c' \neq z$ since z does not lie on a wall. Possibly $c' = c$. Then there is an open ball $B_z(\epsilon)$ of radius $\epsilon > 0$ centered at z so that for any $z' \in B_z(\epsilon)$, $seg[c',z'] \subseteq T$. Since $B_z(\epsilon)$ is reachable from c' and c' is reachable from $c \in Reach$, $B_z(\epsilon) \subseteq Reach$. This contradicts z being on the boundary of $R(c)$ but not on the boundary of any other region.

(2): Suppose z lies on the boundary of $R(c)$, $z \neq c$ and $t(z) = t(c)$. Assume for the sake of contradiction that z is on the boundary of some other region $R(c')$. Note that z is not a corner by our assumption that no two corners have the same time coordinate. By the same assumption, $t(c') \neq t(c)$, and so $t(c') < t(c)$. Thus $speed(c',z) < \infty$ and $z \in Reach$. Let c'' be the last corner on $seg[c',z]$ before z (possibly, $c'' = c'$). Choose $\epsilon > 0$ so that the configurations in $B = B_z(\epsilon) \cap T$ are reachable from c'' (and hence from c') and for $z' \in B$ with $t(z') > t(c)$, $speed(c,z') > \max\{min\_speed(c''), speed(c'',z')\}$. For such z', by the (strict version of the) Triangle Inequality,

$$cost(c'',z') = \max\{min\_speed(c''), speed(c'',z')\} < speed(c,z') \leq cost(c,z').$$

Therefore $z' \in B$ implies either $t(z') \leq t(c)$ and so is not reachable from c or $cost(c,z') > cost(c'',z')$ and so $z' \notin R(c)$. This contradicts z being on the boundary of $R(c)$. □

Corollary 11: If c is a corner and $z \neq c$ is a boundary configuration of $R(c)$ with $t(z) = t(c)$ then z is not reachable.

Proof. Since $t(z) = t(c)$, z is not reachable from c. There are configurations in $R(c)$ arbitrarily close to z because z is on the boundary of $R(c)$. Therefore no region can contain z in its interior. By Lemma 10, z is not in the boundary of any other region. Thus z is not reachable from any other corner. □

Let W_1 and W_2 be the two walls that meet at corner c and the time coordinate of c is greater than the time coordinates of the other endpoints of W_1 and W_2. Consider any two (non-vertical) line segments L_1 and L_2 in T with common endpoint c such that their other endpoints have smaller time coordinates. Then c is said to *block* L_1 and L_2 if there is a line segment L with one endpoint on L_1, the other on L_2 and neither endpoint is c and L intersects W_1 and W_2.

Lemma 12: Let c_1 and c_2 be corners reachable from z_o. If z is a configuration on the

common boundary of $R(c_1)$ and $R(c_2)$ then z is a corner that blocks $seg[c_1,z]$ and $seg[c_2,z]$ or z is on the bisector $b(c_1,c_2)$.

Proof. Without loss of generality assume $t(c_1) < t(c_2)$. The result is clearly true if $z = c_2$ since c_2 is on $b(c_1,c_2)$. Suppose $z \neq c_2$ and z is not a corner blocking $L_1 = seg[c_1,z]$ and $L_2 = seg[c_2,z]$. Let c_3 be the last corner on L_1 before z and c_4 be the last corner on L_2 before z. By Lemma 10, $t(z) > t(c_2)$.

Since z is on the boundaries of both $R(c_1)$ and $R(c_2)$, there is a sequence of configurations $\{x_i\}$ in $R(c_1)$ and a sequence of configurations $\{y_i\}$ in $R(c_2)$ such that $\lim x_i = \lim y_i = z$. Since regions are punctured stars, we may ensure that $t(x_i) \leq t(z)$ and $t(y_i) \leq t(z)$ by replacing, if necessary, each x_i with $seg[c_1,x_i] \cap \{w : t(w) = t(z)\}$ ($\{y_i\}$ is handled similarly).

Suppose c_1, c_2 and z are collinear. Since $seg[c_1,x_i] \subseteq T$ for all i, and $seg[c_2,z] \subseteq T$ there is some i_0 such that, for all $i \geq i_0$, $seg[c_2,x_i] \subseteq T$. Thus $x_i \in R(c_1,c_2)$ for all $i \geq i_0$ and z is in the closure of $R(c_1,c_2)$. By collinearity, $cost(c_1,z) = cost(c_1,c_2)$. But $cost(c_1,c_2) \geq min\_speed(c_2)$ and $cost(c_1,c_2) \geq speed(c_1,c_2) = speed(c_2,z)$. Therefore $cost(c_1,z) \geq cost(c_2,z)$ which implies that $z \in R(c_2,c_1)$ and so $z \in b(c_1,c_2)$.

It is easy to see that $cost(c_3,z) \leq cost(c_1,z)$ using the fact that c_3 is reachable from c_1 and so $min\_speed(c_3) \leq cost(c_1,c_3)$ and the fact that c_1,c_3 and z are collinear. Similarly, $cost(c_4,z) \leq cost(c_2,z)$.

Assume c_1, c_2 and z are not collinear. Since z does not block L_1 and L_2, there is some i_0 such that, for all $i \geq i_0$, $seg[c_4,x_i] \subseteq T$ and $seg[c_3,y_i] \subseteq T$ and so, for all such i, $cost(c_1,x_i) < cost(c_4,x_i)$ and $cost(c_2,y_i) \leq cost(c_3,y_i)$. Therefore $cost(c_1,z) \leq cost(c_4,z) \leq cost(c_2,z)$ and $cost(c_2,z) \leq cost(c_3,z) \leq cost(c_1,z)$. That is,

$$cost(c_1,z) = cost(c_2,z) = cost(c_3,z) = cost(c_4,z).$$

Since $cost(c_1,z) = cost(c_2,z)$ and $t(c_2) > t(c_1)$, $z \in R(c_2,c_1)$. If $c_4 = c_2$ then the above shows that, for $i \geq i_0$, $x_i \in R(c_1,c_2)$. Therefore $z \in b(c_1,c_2)$.

Assume $c_4 \neq c_2$. Suppose $min\_speed(c_2) > speed(c_2,z)$. Then $min\_speed(c_4) \leq cost(c_2,c_4) = min\_speed(c_2)$. Choose i_0 so that, for all $i \geq i_0$, $speed(c_2,y_i) \leq min\_speed(c_2)$, $speed(c_4,y_i) \leq min\_speed(c_2)$ and $seg[c_4,y_i] \subseteq T$. Then $cost(c_2,y_i) = min\_speed(c_2)$ and $cost(c_4,y_i) \leq min\_speed(c_2)$. This, together with $t(c_4) > t(c_2)$, contradicts $y_i \in R(c_2)$. Therefore $min\_speed(c_2) \leq speed(c_2,z)$. In other words, $cost(c_2,z) = speed(c_2,z)$.

Since $cost(c_1,z) = cost(c_2,z)$ either $z \in b(c_1,c_2)$ or $min\_speed(c_1) = min\_speed(c_2)$. Suppose $min\_speed(c_1) = min\_speed(c_2)$. Since $cost(c_1,z) = cost(c_2,z) = speed(c_2,z)$, it follows that $speed(c_1,z) \leq speed(c_2,z)$. Let $\{z_i\}$ be a sequence of configurations on L_1 such that $\lim z_i = z$ and $t(z_i) > t(c_2)$. Then $speed(z_i,z) = speed(c_1,z) \leq speed(c_2,z)$ together with the Triangle Inequality imply $speed(c_2,z_i) > speed(c_2,z) = cost(c_1,z_i)$. Therefore $cost(c_2,z_i) > cost(c_1,z_i)$. Thus $z_i \in R(c_1,c_2)$, $z \in R(c_2,c_1)$ and $\lim z_i = z$, implying $z \in b(c_1,c_2)$. □

Now that we have characterized the types of configurations lying in region boundaries we next show how to decide which corner could be a predecessor of such boundary configurations.

Lemma 13: Let c_1, c_2, $\cdots$, c_m be all the corners such that z lies on the common boundary of $R(c_1)$, $R(c_2)$, $\cdots$, $R(c_m)$. If z is reachable from z_s and $m \geq 1$ then there is a way to split the c_i's different from z into two disjoint sets I_1 and I_2 (one of which may be empty) such that for each $k \in \{1,2\}$, $cost(c_i,z)$ is invariant over $i \in I_k$ and so for some k, $min\_speed(z) = cost(c_i,z)$ for all $i \in I_k$.

Proof. Since z is reachable, $z \in R(c)$ for some corner c. If z were in the interior of $R(c)$, it could not lie on the boundary of any region. Hence z lies on the boundary of $R(c)$ and so c is one of c_1, $\cdots$, c_m. In particular, $min\_speed(z) = cost(c_i,z)$, for some i. Therefore if $m = 1$ we are done.

Suppose $m > 1$. If z is on the bisectors $b(c_i,c_j)$ for all i,j where $1 \leq i < j \leq m$, then

$min\_speed(z) = cost(c_i, z)$ for all i ($1 \leq i \leq m$) such that $z \neq c_i$ and every $c_i \neq z$ is a predecessor of z. Suppose z is not on the bisector $b(c_i, c_j)$ for some $c_i, c_j \neq z$. Let $L_k = seg[c_k, z]$, $1 \leq k \leq m$. Then z must block L_i and L_j by Lemma 12. Let I_1 be the set of all $c_k \neq z$ such that z blocks L_j and L_k. Similarly let I_2 be the set of all $c_k \neq z$ such that z blocks L_i and L_k. Notice that, if c_k and c_l are both in I_1 (or both in I_2), then z does not block L_k and L_l. Hence the result follows. □

Lemma 13 in effect states that, for a configuration z on the boundary of one or more regions, knowing the regions adjacent to z and the walls that contain z (if z is a corner) is enough to be able to compute a predecessor of z and $min\_speed(z)$ (provided that the $min\_speed(c)$ has been computed for corners c with $t(c) < t(z)$). If z is not a corner, knowledge of all regions meeting at z is sufficient.

Lemma 14: For a corner c, configuration $z \in R(c)$, and $\epsilon > 0$, let $r(c, z, \epsilon) = R(c) \cap B_z(\epsilon) \cap \{z' \mid t(z') > t(z)\}$, $B_z(\epsilon)$ being the open ϵ-ball around z. If $r(c, z, \epsilon)$ is 1-dimensional, then it lies along a wall or each point of it lies on the common boundary of at least two distinct regions $R(c_1)$ and $R(c_2)$ where $c_i \neq c$. In other words, locally one-dimensional portions of a region follow walls or are confined between two *distinct* two-dimensional regions.

Proof. Suppose $r(c, z, \epsilon)$ is 1-dimensional. Since $R(c)$ is a punctured star with respect to c, $r(c, z, \epsilon)$ must be a segment of a line through c. Suppose $r(c, z, \epsilon)$ does not lie along any wall. *Reach* has no 1-dimensional pieces, so $r(c, z, \epsilon)$ must lie in the interior of *Reach*. Choose any $z' \in r(c, z, \epsilon)$. Then there are regions $R(c_1)$ and $R(c_2)$ on opposite sides of $r(c, z, \epsilon)$ with z' in their common boundary. If $c_1 = c_2$ then since $R(c_1)$ must be a punctured star, c_1 must lie on the line containing $r(c, z, \epsilon)$. Clearly $t(c_1) < t(c)$, for otherwise the configurations in $r(c, z, \epsilon)$ would be in $R(c_1, c)$ and thus certainly not in $R(c)$. However, line segments from c_1 to configurations arbitrarily close to z' on (at least) one side of $r(c, z, \epsilon)$ must penetrate at least one of the walls that meet at c. Thus $c_1 \neq c_2$. □

The lemmas in this section have shown that a configuration z in *Reach* is either in the interior of some region, on a wall (or such that $t(z) = t(c)$ for some corner c) in the boundary of some region or on the common boundary of two or more regions. By definition any configuration z in the interior of region $R(c)$ is such that c is a predecessor of z. Lemma 13 indicates how to find a predecessor of z if z is on the common boundary of several regions. If z is a reachable configuration on a wall and on the boundary of a single region $R(c)$, then $z \in R(c)$. Therefore the boundaries of *Reach* together with the partition of *Reach* into labeled cells, where each cell is (a connected component of) the nonempty interior of a region and is labeled by c if the region is $R(c)$, provide sufficient information to determine a predecessor for any query configuration.

4.2. Planar partition

We wish to define a planar partition whose faces are the connected components of region interiors or open connected subsets of *Illegal*. Let $F_i(c)$ denote a connected component of the interior of region $R(c)$. We define the following to be *arcs* of the desired planar map:

(i) the vertical line segments $\{z : t(z) = t(z_s) \text{ and } x(z) \geq x(z_s)\}$ denoted by V_1 and $\{z : t(z) = t(z_s) \text{ and } x(z) \leq x(z_s)\}$ denoted by V_2

(ii) for each pair of faces $F_i(c_1)$ and $F_j(c_2)$, each maximal, nontrivial (i.e. not a single point), time-monotonic, connected portion of the bisector $b(c_1, c_2)$ on the common boundary of $F_i(c_1)$ and $F_j(c_2)$

(iii) for each face $F_i(c)$ and wall w, a maximal, nontrivial, connected section of w in the boundary of $F_i(c)$

(iv) for each face $F_i(c)$, the nontrivial segment (or ray) going vertically upward or downward from c in the boundary of $F_i(c)$.

The *faces* of the map are the connected components of the complement of the union of the arcs in the plane. Each face is either completely contained in *Reach* or completely contained in

Illegal. Each face in *Reach* is some $F_i(c)$. A face in *Illegal* is called an *illegal face*. The *edges* of the map are defined to be the maximal connected sections of the arcs of the map that contain intersections with other arcs only at the endpoints. Edges are labeled according to the owners of their incident regions and the wall or the vertical segment that contains them (if any). We say that an edge is a bisector edge, a wall edge or a vertical edge if it is a section of a type (ii), (iii) or (iv) arc, respectively. Bisector and wall edges are directed away from the endpoint with the smaller time coordinate. The *vertices* of the map are the endpoints of the edges together with the starting configuration z_o. Notice that all configurations on edges are reachable except those on a vertical edge (not including the corner from which it starts). Also, Lemma 10 implies that no arc can intersect the interior of a type (iv) arc.

Figure 3 is an example of such a planar partition where the region of each corner consists of at most one face. The solid lines are the walls of the time obstacles. The dotted arcs are the bisector edges separating two regions and the dashed lines are V_1, V_2 and the vertical edges. Notice that the bisector edge emanating from corner c_6 and separating $R(c_3)$ from $R(c_6)$ is a section of the bisector $b(c_3, c_6)$ and it begins as a hyperbolic curve before becoming a ray.

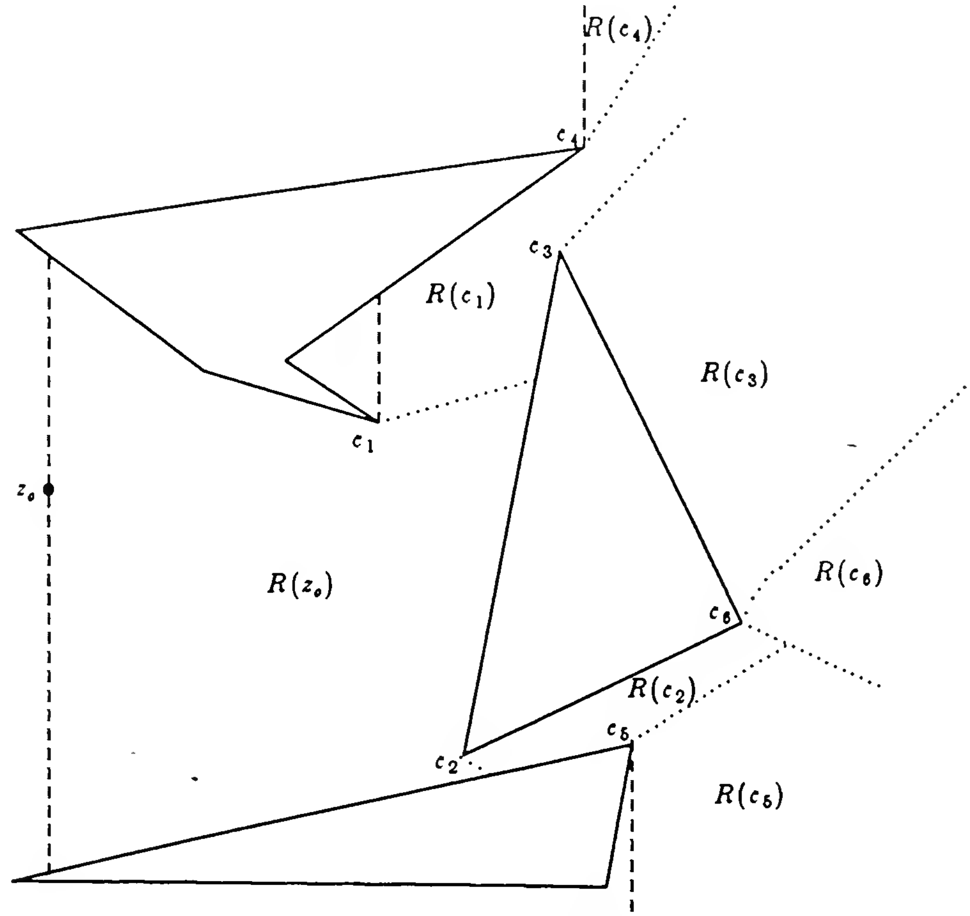


Figure 3. Planar partition

We use a sweep-line approach for computing this labeled planar map. During the sweeping we wish to determine not only the map but also, for each corner c in *Reach*, *min_speed*(c) and a predecessor of c. The sweeping is performed by a vertical line l_t moving in the direction of increasing time. The information necessary to compute the map is provided by the order (with respect to the x-coordinate) of intersections of l_t with the edges of the map, the labels of

such edges and the labels of faces lying between consecutive intersections. This order of the intersection of the edges with l_t changes when edges meet (i.e. when l_t encounters vertices of the map).

Suppose v is a vertex of the map and let $c_1, \cdots, c_m$ be all corners (except for v itself) on the boundary of whose regions v lies, ordered top-to-bottom according to the direction of $seg[c_i, v]$. Since the interiors of regions are punctured stars there are a constant number of faces in a neighborhood (with respect to $T_v = \{z \in T : t(z) \geq t(v)\}$) about v, namely at most one connected component each of the interiors of $R(c_1)$ and $R(c_m)$ (two if $c_1 = c_m$), and at most two of $R(v)$ if v is a corner. If v is a corner such that v is the earliest endpoint of both of the walls W_1 and W_2 that meet at v and $c_1 = c_m$, then the connected component of the interior of $R(c_1)$ near v may be separated into two pieces after v by W_1 and W_2. (Figure 4 illustrates the situation where v is incident to four faces, all of which meet T_v.)

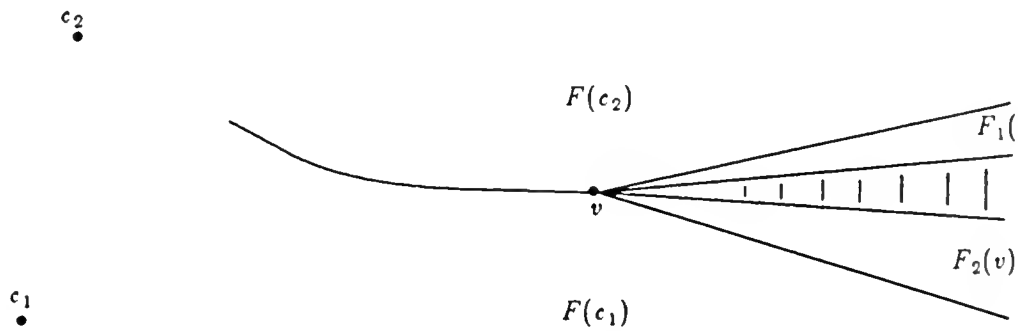


Figure 4. Four faces incident to corner v:

$$min\_speed(c_1) = min\_speed(c_2) = speed(c_1, v) = speed(c_2, v).$$

Using the slopes of $seg[c_i, v]$, of the bisector $b(c_1, c_m)$ (and $b(c_1, v)$ and $b(c_m, v)$ if v is a corner), and of the wall(s) (if any) on which v lies, it is straight forward to determine which of the four possible regions have nonempty interiors in a neighborhood of v (with respect to T_v) and what their boundaries are. Therefore the order (and labels) of the intersections of the edges with the sweep-line $l_{t(v)+\epsilon}$ (for appropriately small ϵ) as well as the labels of the regions between the intersections can be determined in time proportional to the degree of v.

Lemma 15: There are $O(n)$ illegal faces and each is simply connected.

Proof. One illegal face H consists of all configurations with time component less than $t(z_o)$ since no such configuration can be unreachable and $V_1 \cup V_2$ separates H from all other illegal configurations. Clearly H is simply connected.

Let H' be any other illegal face. Suppose H' is not simply connected. Then there is a simple loop $C \subseteq H'$ surrounding some point $z \in Reach$. (An illegal face that is not simply connected must enclose some reachable point for no arc besides V_1 and V_2 is incident to more than one illegal face.) But $z \in Reach$ implies that there is a path p in $Reach$ from z_o to z. Therefore z_o must also be surrounded by C (otherwise p would have to intersect C contradicting $C \subseteq H' \subseteq Illegal$). However this implies that $V_1 \cup V_2$ intersects C contradicting the fact that face H' contains no configurations on an arc. Thus H' must be simply connected.

A simple sweep-line argument similar to that found in Theorem 3.1 of [RS] can be used to show that there are $O(n)$ illegal faces occurring after the arcs V_1 and V_2 since some configuration with earliest time coordinate on the boundary of such an illegal face is a corner. As there is only one illegal face before V_1 and V_2, there are a total of $O(n)$ illegal faces. □

Since the interiors of regions are punctured stars, if there is no corner with time component t, no new faces $F_i(c)$ appear as the sweep-line passes through t. The sweep-line argument in the proof of Lemma 15 shows that no illegal faces can begin either. If v is a corner then

a constant number of new faces may begin at v. Therefore the total number of faces in the map is $O(n)$ and so there are $O(n)$ vertices at which three or more edges meet. The only possible vertices of degree two are corners (including z_o) and endpoints of vertical segments from a corner and so there are $O(n)$ such vertices. Thus there are a total of $O(n)$ vertices, faces and edges in the map. The full description of the map has linear size since each edge has constant complexity.

A predecessor for corner c and $min\_speed(c)$ are computed when the sweep-line reaches $t(c)$. We assume inductively that for each corner c' such that $t(c') < t(c)$ we have computed $min\_speed(c')$ and a predecessor for c'. By Lemma 13 a predecessor of c (and hence $min\_speed(c)$) can be determined in time proportional to the number of edges meeting at c given the order of the intersections of the edges with the sweep-line just before $t(c)$.

In order for the sweep-line method to work it is necessary that each vertex of the map can be determined using only the knowledge about the part of the map that occurs no later than the vertex. Therefore we show the following lemma.

Lemma 16: If v is a non-corner vertex of the labeled planar map, it has out-degree 1 and in-degree at least 2. (For the purposes of this discussion a vertical edge is considered outgoing at the corner from which it emanates.)

Proof. If v is on a type (ii) or type (iii) arc then by Lemma 14, it is not possible for v to have total degree 1. If v is on some vertical segment from a corner then it must be the intersection of a type (iv) arc and a wall and so the result holds. Therefore, since v is not a corner, v must be the intersection of at least one bisector and a wall or the intersection of two or more bisectors. If v is the intersection of bisectors and a wall then the fact that the interior of regions are punctured stars implies that the bisector edges with v as one endpoint must have their other endpoint earlier in time. Thus the only edge leaving v is a wall edge.

Suppose v is the intersection of two or more bisector edges and no wall edges with $out\_degree(v) > 1$. Then the fact that the faces between these bisector edges are punctured stars implies that v must be a corner, contradicting our assumption. Similarly, the absence of an outgoing edge leads to a contradiction. Suppose $in\_degree(v) = out\_degree(v) = 1$. Let $b(c_1, c_2)$ and $b(c_3, c_4)$ be the two distinct bisectors along which the two edges lie. By definition of bisector edges, the two edges incident to v must (locally near v) separate $F(c_1)$ from $F(c_2)$ and $F(c_3)$ from $F(c_4)$ where $F(c_i) \subseteq R(c_i)$ is some face in the map. But by definition of a face and by assumption that $in\_degree(v) = out\_degree(v) = 1$, we must have $F(c_1) = F(c_3)$ and $F(c_2) = F(c_4)$ (or $F(c_1) = F(c_4)$ and $F(c_2) = F(c_3)$). This implies that v is in the relative interior of an edge contradicting the assumption that v is a vertex. □

4.3. The algorithm

The results of the previous section outline the method we use to compute the desired labeled planar map so that queries about minimum speed paths can be quickly answered. We now state how to implement the indicated procedures to provide $O(n\log n)$ preprocessing time. Let S be the list (sorted by increasing x-coordinate) of labels of map edges that intersect the sweep-line and of the labels of the faces lying between consecutive intersections. At each point in time as the sweep-line proceeds, Q will be a sorted by time coordinate list of potential vertices (called events) of the planar map. The steps of the algorithm are as follows.

1. Sort corners c with $t(c) > t(z_o)$ by increasing time coordinate, place them in Q and set $min\_speed(z_o) = 0$

2. Let w_1 be the wall (if one exists) which the upward vertical line from z_o first intersects. Place into S the edge label for the portion of w_1 in the boundary of $R(z_o)$. Do a similar operation for the first wall that the downward line from z_o intersects.

3. Remove from Q the earliest vertex v in Q. Remove from S all edges and faces that come to an end at v and from Q all events involving these edges. If v is a corner set $pred(v)$ and $min\_speed(v)$ according to Lemma 13. Determine the edges with v as their earlier endpoint and their initial slopes as indicated in the previous discussion so that their order (by

NYU COMPSCI TR-389 c.1
Aronov, Boris
Minimum speed motions